Genesys Mobile Services Deployment Guide

Cassandra Security

12/19/2025

Contents

- 1 Cassandra Security
  - 1.1 Protecting Data Stored in the Cassandra Database
  - 1.2 Cassandra Authentication
  - 1.3 Cassandra Gossip TLS
  - 1.4 Cassandra JMX TLS

Cassandra Security

This page discusses security configurations for Cassandra.

Protecting Data Stored in the Cassandra Database

The *<Genesys Mobile Services installation directory>/etc/cassandra.yaml* file enables or disables encryption of Cassandra inter-node communication using TLS_RSA_WITH_AES_128_CBC_SHA as the cipher suite for authentication, key exchange, and encryption of the actual data transfers. To encrypt all inter-node communications, set to *all*. You must also generate keys, and provide the appropriate key and trust store locations and passwords. Details about Cassandra options are available from:

- internode_encryption
- authenticator

All transient service session-related data is stored in a Cassandra database that uses memory and the file system. See the *<Genesys Mobile Services installation directory>/data* folder. Files located here should be protected from unauthorized access.

Cassandra Authentication

The following steps show how to protect access to the Cassandra `gsg` keyspace. All of the files and options noted below are in the GMS installation package.

1. Before editing the following files, stop the GMS node.
   - To comment a line, add a # at the beginning of the line.
   - To uncomment a line, delete the # from the beginning of the line.
2. When you have finished editing and saving the files, restart the GMS node.

cassandra.yaml file

1. Open the *<Genesys Mobile Services installation directory>/etc/cassandra.yaml* file.
   - Comment the line: `authenticator: org.apache.cassandra.auth.AllowAllAuthenticator`.
   - Uncomment the line: `#authenticator: com.genesyslab.gsg.storage.auth.SimpleAuthenticator`.
   - Comment the line: `authority: org.apache.cassandra.auth.AllowAllAuthority`.
   - Uncomment the line: `#authority: com.genesyslab.gsg.storage.auth.SimpleAuthority`.
2. Save and close the file.

access.properties file

1. Open the *<Genesys Mobile Services installation directory>/etc/access.properties* file.
   - Uncomment the line: `#<modify-keyspaces>=cassandra`. Note that the value, `cassandra`, is the user login that will be used in the GMS server node (server-side and client-side). You must change this user login. However, for the examples shown here, `cassandra` will continue to be used as the user login. This line tells the Cassandra server that the `cassandra` user can modify keyspaces, which is needed in order to create the `gsg` keyspace.
   - Uncomment the line: `#gsg.<rw>=cassandra`. This grants read/write permissions to the `cassandra` user login to be able to read and write into the `gsg` keyspace.
2. Save and close the file.

passwd.properties file

1. Open the *<Genesys Mobile Services installation directory>etc/passwd.properties* file.
2. Uncomment one of the following lines:
   - `#cassandra=pelops` if you want to use a PLAIN text password in this file. Make sure that you also change the user login and password.
   - `#cassandra=b57694b2c9cfc6fbaf00a7033b2a7e4c` if you want to use an MD5 password in this file. Make sure that you also change the user login and password. You can use any MD5 tools to generate the MD5 result of your password, for example, using md5sum:

```
$ echo -n "pelops" | md5sum

b57694b2c9cfc6fbaf00a7033b2a7e4c
```

3. Save and close the file.

launcher.xml

1. Open the *launcher.xml* file.
2. Locate the parameter `cassandra.login`.
3. In the `<format type="string" default=""/>` line within this parameter, change the default by using your cassandra user login previously defined. For example, `<format type="string" default="cassandra"/>`.
4. Locate the parameter `cassandra.password` .
5. Change the following line in this parameter: `<format type="string" default=""/>` using the password defined for your cassandra user login. This password must be in PLAIN text (even if you used MD5 for hashing your password in the previous steps). For example, `<format type="string" default="pelops"/>`.
6. Locate the parameter `cassandrapasswordmode` and change the mode according to the way you recorded your password in the `passwd.properties` file:
   - `<format type="string" default="-Dpasswd.mode=MD5" />` for a MD5 password or
   - `<format type="string" default="-Dpasswd.mode=PLAIN" />` if your password was recorded in plain text.

7. Save and close the file.

The following shows an example of the parameters in the launcher.xml file:

```
<parameter name="cassandralogin" displayName="cassandra.login" mandatory="false">
<description><![CDATA[ Cassandra Server Login for Client]]></description>
<valid-description><![CDATA[]]></valid-description>
<effective-description/>
<format type="string" default="user"/>
<validation>
</validation>
</parameter>

<parameter name="cassandrapassword" displayName="cassandra.password" mandatory="false">
<description><![CDATA[ Cassandra Server Password for Client]]></description>
<valid-description><![CDATA[]]></valid-description>
<effective-description/>
<format type="string" default="password "/>
<validation>
</validation>
</parameter>



<parameter name="password" displayName="cassandrapassword" mandatory="true" hidden="true"
readOnly="true">

<description><![CDATA[Security: cassandra password file]]></description>
<valid-description><![CDATA[]]></valid-description>
<effective-description/>
<format type="string" default="-Dpasswd.properties=./etc/passwd.properties" />
<validation></validation>
</parameter>
<parameter name="access" displayName="cassandraaccess" mandatory="true" hidden="true"
readOnly="true">
<description><![CDATA[Security: cassandra access file]]></description>
<valid-description><![CDATA[]]></valid-description>
<effective-description/>
<format type="string" default="-Daccess.properties=./etc/access.properties" />
<validation></validation>
</parameter>
<parameter name="passwdmode" displayName="cassandrapasswordmode" mandatory="true"
hidden="true" readOnly="true">
<description><![CDATA[Security: cassandra password mode]]></description>
<valid-description><![CDATA[]]></valid-description>
<effective-description/>
<format type="string" default="-Dpasswd.mode=MD5" /> # or PLAIN
<validation></validation>
</parameter>
```

Limitation: If the password in GMS is no longer required, you must undo all of the changes in the files.

Cassandra Gossip TLS

In Cassandra version 1.1.x (the Cassandra version in GMS), internode (gossip) encryption is set up in the *cassandra.yaml* file. Locate the following lines in the *cassandra.yaml* file:

```
encryption_options:
 internode_encryption: none
 keystore: conf/.keystore
 keystore_password: cassandra
 truststore: conf/.truststore
 truststore_password: cassandra
```

Replace `internode_encryption: none` with `internode_encryption: all`, as shown in the following example:

```
encryption_options:
 internode_encryption: all
 keystore: conf/.keystore
 keystore_password: cassandra
 truststore: conf/.truststore
 truststore_password: cassandra
```

For managing keystore and truststore (and password), see the Oracle documentation keytool-Key and Certificate Management Tool or the Oracle security guide.

Cassandra JMX TLS

Cassandra monitoring and management can be done using a Java Management Extensions (JMX) tool. The JMX access must be protected in order to avoid any remote managing on the GMS embedded Cassandra. To protect JMX access, edit the `launcher.xml` file that contains the following lines (by default):

```
<parameter name="jmxport" displayName="jmxport" mandatory="true" hidden="true"
readOnly="true">
    <description><![CDATA[JMX related]]></description>
    <valid-description><![CDATA[]]></valid-description>
    <effective-description/>
    <format type="string" default="-Dcom.sun.management.jmxremote.port=9192" />
    <validation></validation>
</parameter>
<parameter name="jmxssl" displayName="jmxssl" mandatory="true" hidden="true" readOnly="true">
    <description><![CDATA[virtual machine related]]></description>
    <valid-description><![CDATA[]]></valid-description>
    <effective-description/>
    <format type="string" default="-Dcom.sun.management.jmxremote.ssl=false" />
    <validation></validation>
</parameter>
<parameter name="jmxauthenticate" displayName="jmxauthenticate" mandatory="true"
hidden="true" readOnly="true">
    <description><![CDATA[virtual machine related]]></description>
    <valid-description><![CDATA[]]></valid-description>
    <effective-description/>
    <format type="string" default="-Dcom.sun.management.jmxremote.authenticate=false" />
    <validation></validation>
</parameter>
```

By default, the TLS and authentication parameters are disabled:

```
   com.sun.management.jmxremote.ssl=false
   com.sun.management.jmxremote.authenticate=false
```

For information about enabling these parameters and managing JMX and TLS, see the *Monitoring and*

Management Using JMX Technology chapter in the Oracle Java SE Monitoring and Management Guide.